

Практическое занятие № 2

Группировка данных и агрегатные функции

Цель занятия. Научиться выполнять группировку данных и вычислять агрегатные функции в операторе запроса в реляционной БД.

Группировка результатов выборки из БД - это мощная функция SQL, которая позволяет вычислять ключевые статистические функции для группы записей (например, средние, максимальные и минимальные значения в группе записей).

Обычная выборка может вернуть только строки. С помощью функции группирования можно получать группы записей и вычислять показатели для каждой группы. В данной работе вы узнаете, как группировка делает SQL-запросы намного более мощными и разнообразными. Но вначале рассмотрим агрегатные функции.

В примерах и упражнениях будем использовать БД Поставки (рис.1). На продуктовую оптовую базу поступает продукция от поставщиков. Данные таблиц БД приведены в [практическом занятии 1](#).



Рис. 1 База данных Поставки

Агрегатные функции

Агрегатные функции (в стандарте SQL они называются функциями над множествами) используются для обобщения отдельных значений данных. Агрегатные функции вычисляют некоторое интегральное значение для заданного множества строк. Таким множеством строк может быть или вся таблица или группа строк, если агрегатная функция применяется к сгруппированной таблице.

При вычислении агрегатной функции по некоторой колонке не учитываются неопределенные значения (NULL). Если в агрегатной функции используется ключевое слово DISTINCT, то из процесса вычислений исключаются значения-дубликаты. Если агрегатная функция применяется без ключевого слова DISTINCT, то вычисление производится над всеми значениями.

В стандарте SQL определены пять агрегатных функций:

- 1) COUNT - функция подсчета количества значений. Применяется в двух форматах:

COUNT()* подсчитывает число строк в заданном множестве. Учитываются все строки - в том числе одинаковые и содержащие значение null.

COUNT([DISTINCT] колонка) подсчитывает число не Null-значений в заданном столбце (с DISTINCT – количество разных значений).

- 2) MAX(выражение) - определяет максимальное значение.
- 3) MIN(выражение) - определяет минимальное значение.
- 4) SUM(выражение) - вычисляет сумму значений.
- 5) AVG(выражение) - вычисляет среднее значение.

Последние 2 функции используются только для числовых значений.

Аргументом агрегатной функции может быть как отдельная колонка, так и выражение

Например, SUM(KOL*CENA)

Пример 1. Вывести общий объем всех поставок. Функция вычисляется по всей таблице:

```
SELECT SUM(kol) from kart
```

Результат запроса:

SUM(kart.kol)	
1	201

Пример 2. Определить количество строк, в которых заполнена колонка контактов в таблице поставщиков

```
SELECT COUNT(*) всего, COUNT(contacts) заполнено from spost
```

Результат запроса:

всего		заполнено	
1	8		6

В запросе первая агрегатная функция COUNT(*) вычисляет общее количество строк в таблице, а вторая COUNT(contacts) – количество строк, в которых колонка контактов не имеет значение null (т.е. заполнена).

Пример 3. Определить количество разных наименований продукции, поставленной поставщиком с кодом 1

```
SELECT COUNT (DISTINCT prod_kod) "кол-во"  
FROM book b, kart k  
WHERE b.book_id=k.book_id and post_kod=1
```

Результат запроса:

кол-во	
1	2

Упражнение 1.

- 1) Определить максимальный объем поставки в натуральном выражении.
- 2) Определить среднюю цену поставки.
- 3) Определить количество поставок в январе 2023 г.

Раздел GROUP BY

Раздел GROUP BY позволяет сгруппировать отобранные данные по одной или нескольким колонкам и вычислить для каждой группы агрегатную функцию

GROUP BY <список колонок>

Результатом будет "*сгруппированная таблица*".

Важно отметить, что в список выборки оператора SELECT, содержащего раздел GROUP BY, можно включать *только* агрегатные функции и колонки, *которые входят в раздел группировки*.

Пример 4. 1) Получить суммарные поставки по продукции для поставщика с кодом 1

```
SELECT prod_kod, sum(kol) "общий объем"  
FROM book b, kart k  
WHERE b.book_id=k.book_id  
and post_kod=1  
GROUP BY prod_kod
```

Алгоритм выполнения запроса:

- 1) Будет выполнено прямое произведение таблиц book и kart.
- 2) К полученным строкам будет применен раздел WHERE, условия которого выполняют естественное соединение таблиц и отбирают строки (продажи) для поставщика с кодом 1.
- 3) Полученные строки будут сгруппированы так, чтобы в каждую группу попадали строки с одинаковыми значениями в колонке prod_kod.

post_kod	kart_id	prod_kod	cena	kol	book_id
1	1	1	45.0	10	1
1	8	2	55.0	14	4
1	2	2	52.0	16	1

4) В каждой группе будут просуммированы значения в колонке kol.

5) От каждой группы в результирующую таблицу будет включена одна строка.

Результат запроса:

prod_kod	общий объем
1	10
2	30

Пример 5. Получить суммарные поставки в стоимостном выражении по поставщикам

```
SELECT sp.post_name, sum(kol*cena) as stoim
FROM spost sp, book b, kart k
WHERE sp.post_kod =b.post_kod
      and b.book_id=k.book_id
GROUP BY sp.post_name
```

Результат запроса:

post_name	stoim
1 АО Автолог	1,266.0
2 АО Продлог	2,052.0
3 ОАО Черемушки	850.0
4 ООО Контакт	4,132.0
5 ООО Логистик	745.0
6 ООО Москворечье	569.0

Упражнение 2.

- 1) Определить суммарные объемы поставок по каждой продукции
- 2) Определить количество разных наименований продукции по каждому поставщику.

Раздел HAVING

Условия отбора можно применять не только к строкам таблиц, но и к группам. Раздел HAVING содержит условное выражение, вычисляемое для каждой группы, определенной в разделе GROUP BY. Это условное выражение должно содержать агрегатные функции, вычисляемые для каждой группы.

Рассмотрим пример. Выбрать продукцию, для которой максимальный объем поставки больше 10.

Первый вариант запроса, который напрашивается будет таким:

```
SELECT prod_kod FROM kart WHERE MAX(kol)>1000
```

Но данный оператор запроса не верен, т.к. условие WHERE вычисляется для каждой строки BOOK, т.е. до того как будет вычислена агрегатная функция, которая применима только к группе строк.

Для отбора групп вместо раздела WHERE используется раздел HAVING. Этот раздел задает условие отбора для сгруппированной таблицы.

HAVING <условие отбора>

В условии отбора раздела HAVING используют те же логические операции (предикаты), что и в условии отбора раздела WHERE. Но в условии отбора раздела HAVING колонки можно использовать в качестве аргумента агрегатных функций, вычисляющих в данном случае некоторое агрегатное значение для всей группы строк.

Результатом выполнения раздела HAVING будут те строки сгруппированной таблицы, для которых условия отбора дает true.

Замечание. В одном запросе могут встретиться как условия отбора строк в разделе WHERE, так и условия отбора групп в разделе HAVING. Разница в том, что условие раздела WHERE применяется к выражениям, которые можно вычислить по каждой строке, а в разделе HAVING - к агрегатным функциям по группам строк.

Правильная запись оператора запроса для приведенного выше примера:

```
SELECT prod_kod FROM kart
GROUP BY prod_kod
HAVING MAX(KOL)>10
```

Пример 6. Получить суммарные поставки по видам продукции, для которых общий объем поставки превышает 20

```
SELECT PROD_KOD, SUM(KOL) as SUM_KOL
FROM kart
GROUP BY prod_kod
HAVING sum(kol) > 20
```

Результат запроса:

	prod_kod	SUM_KOL
1	1	28
2	2	53
3	5	32
4	6	50

Пример 7. Получить наименования продукции, которая поставлялась более одного раза

```
SELECT prod_name, count(*)
FROM sproid sp, kart k
WHERE k.prod_kod =sp.prod_kod
GROUP BY sp.prod_name
```

HAVING count(k.prod_kod) > 1

Результат запроса:

	prod_name	кол.поставок
1	Батон нарезной	3
2	Молоко 1.8%	2
3	Молоко 3.5%	2
4	Печенье сдобное	4
5	Сыр Швейцрский	2
6	Хлеб ржаной	2

Упражнение 3.

- 1) Вывести поставщиков, у которых общий объем поставок в стоимостном выражении превышает 1000
- 2) Вывести продукцию, у которой минимальная поставка превышает 10

Задание

1. Выполните упражнения на учебной БД
2. Составьте похожие запросы для вашей БД